The Majority Wins: a Method for Combining Speaker Diarization Systems

Marijn Huijbregts[1], *David van Leeuwen*[2,3] *and Franciska de Jong*[1]

[1]University of Twente, Department of Electrical Engineering, Mathematics and Computer Science
[2]TNO Human Factors, [3]Radboud University, department of Language and Speech
{huijbreg,fdejong}@ewi.utwente.nl, david.vanleeuwen@tno.nl

Abstract

In this paper we present a method for combining multiple diarization systems into one single system by applying a majority voting scheme. The voting scheme selects the best segmentation purely on basis of the output of each system. On our development set of NIST Rich Transcription evaluation meetings the voting method improves our system on all evaluation conditions. For the single distant microphone condition, DER performance improved by 7.8% (relative) compared to the best input system. For the multiple distant microphone condition the improvement is 3.6%.

Index Terms: Speaker diarization

1. Introduction

The goal of speaker diarization is to automatically segment an audio recording into speaker homogeneous regions. When the identity of each speaker is not known and even the number of speakers is unknown, it is the task of a diarization system to anonymously label each speaker in the recording and answer the question: 'Who spoke when?' [1].

Since 2004 NIST has organized evaluations of speaker diarization technology on the meeting domain [2]. At each benchmark, diarization systems are evaluated, for a number of audio recording conditions. The primary evaluation condition allows the use of audio recorded from multiple distant microphones. As an optional task, NIST also evaluates the performance of diarization systems for the condition in which the audio input comes from just a single (distant) microphone.

During the development of our speaker diarization system for the NIST RT09 evaluation, the AMI RT09 system, we noticed remarkable variation in performance on individual recordings each time we made a relatively small adjustment to our system. Although because of such adjustments the overall diarization error rate did not change much, the error rate of individual meetings could easily change with more than twenty percent.

These variations in diarization error rate occur because one single clustering mistake can be responsible for a large part of the error. When the system misses one speaker or divides the speech of one speaker over two clusters, the error rate will increase significantly. Adjusting the system even a little might cause the system to make one of these mistakes on one of the recordings. Therefore, instead of picking the best (overall) performing system configuration during fine-tuning experiments, we investigated the possibility of combining promising system configurations into one single system. If we are able to combine systems in a robust manner, the output will be less sensitive to the system configuration parameters.

In this paper we will discuss our first attempt to combine speaker diarization systems on the basis of a voting system. The idea of this method originates from the Recognizer Output Voting Error Reduction (ROVER) method that is being used in automatic speech recognition [3]. Before discussing our voting algorithm in section 4, we will first briefly discuss our speaker diarization system in section 2. In section 3 we will describe the small tuning adjustments that we have made to the system and list the achieved diarization error rates of these individual system set-ups. In section 5 we will apply our voting algorithm on the various system configurations and we will conclude this paper with a discussion in section 6.

2. The speaker diarization system

Our speaker diarization system is based on a system originally described in [4]. The system consists of three main components: feature extraction, speech activity detection and speaker diarization. An extensive description of these components can be found in [5, 6]. In this section we will only provide a short description of these components so that we can easily explain how we performed our tuning experiments in section 5.

2.1. Feature Extraction

The meetings under evaluation are recorded with multiple distant microphones. The audio signal of each microphone is first passed through a Wiener filter for noise reduction. We used the Wiener filtering application from the Aurora 2 front-end [7]. After Wiener filtering, the channels are combined into one 'enhanced' channel using delay and sum beamforming software (BeamformIt 2.0[1]). This software determines the delay of each signal relative to the other signals and removes this delay before summing all signals together [8]. From the resulting 16kHz audio file, the first nineteen Mel Frequency Cepstral Coefficients (MFCC) are extracted.

2.2. Speech activity detection

For RT07s we developed, in collaboration with ICSI, a robust speech activity detection (SAD) component that is described in [5]. This components finds all speech regions in two steps: first, using a bootstrapping speech/non-speech detection an initial segmentation is created and models for speech, silence and audible non-speech are generated. In the second step these models are applied to generate the final speech/non-speech segmentation. The original speech activity detection component uses MFCC features without Cepstrum Variance Normalization (CVN). In one of the tuning experiments in section 3, we have added CVN to the SAD component.

[1]www.icsi.berkeley.edu/~xanguera/beamformit

2.3. Speaker diarization

The diarization component that uses the speech segments provided by the SAD component as input, is based on the use of Hidden Markov Models (HMM) with Gaussian Mixture Models (GMM) as probability density functions and is described in-depth in [6]. In this system, each speaker is represented by a string of states that share a single GMM. Initially a high number of strings is placed in parallel in the HMM and by using agglomerative clustering, the number of strings is reduced until the correct number of speakers is reached. The final speaker segmentation is obtained by performing a Viterbi search on all audio that contains speech. All audio that is processed by the same string of states during this alignment is grouped together as speech from one speaker. By using a string of states to represent each speaker (instead of a single state), a minimum duration of each speech segment is guaranteed.

A by-product of the beam forming toolkit are the actual delays between microphones with which a sound is recorded. In previous evaluations it has been shown that these delays can be applied as a second feature stream [9]. In our new system we have implemented the use of delay features in a second feature stream. In section 3 we will describe some tuning experiments that we performed to find out what window length can best be used to calculate the delay features for use in our system.

3. Fine-tuning the system

During the development of our speaker diarization system we have created numerous variations in system configuration in order to fine-tune the system. In this section we will describe a number of these system configurations and show the performance of these configurations on our development set for RT09. In section 5 we will use the individual system configurations to create one combined system output. In table 1 we have listed the meetings that we used as development set.

AMI20041210-1052, AMI20050204-1206, CMU20050228-1615
CMU20050301-1415, CMU20050912-0900, CMU20050914-0900
CMU20061115-1030, CMU20061115-1530, EDI20050216-1051
EDI20050218-0900, EDI20061113-1500, EDI20061114-1500
ICSI20000807-1000, ICSI20010208-1430, NIST20030623-1409
NIST20030925-1517, NIST20051024-0930, NIST20051102-1323
NIST20051104-1515, NIST20060216-1347, TNO20041103-1130
VT20050304-1300, VT20050318-1430, VT20050408-1500
VT20050425-1000, VT20050623-1400, VT20051027-1400

Table 1: *The* 27 *conference meetings that we used as test set*

3.1. Speech activity detection

The SAD system, developed for RT07s [5], does not apply cepstrum variance normalization (CVN) or cepstrum variance flooring. We added CVN and variance flooring to the SAD component and initial SAD experiments indicated that these adjustments improve the component. Next, we wanted to know if our diarization system would improve as well using the new SAD component. Therefore, we performed a diarization experiment for both the Single Distant Microphone (SDM) and Multiple Distant Microphone (MDM) conditions.

We also performed another set of experiments concerning the SAD component. In [10] it was shown that noise reduction, especially in combination with beam-forming of the audio channels, improves the diarization performance considerably, but it wasn't tested if noise reduction improves the performance of the SAD component. We have adopted the use of noise reduction for diarization, but for the SAD component we experimented with a configuration without noise reduction.

In table 2 the results of these two sets of experiments are listed for both the SDM and MDM conditions. In the remainder of this paper we will refer to the original SAD RT07s SAD system as SAD_{org}, we will refer to the SAD configuration with CVN and cepstrum flooring as SAD_{cvn} and to the SAD configuration without noise reduction as SAD_{noise}.

The diarization experiments in table 2 show that both new SAD configurations decrease the diarization performance. Note that the missed speech and false alarms are indeed low for SAD_{cvn}, but that the speaker error is high for that configuration. For SDM the SAD_{noise} performance is comparable to SAD_{org}, but for MDM it is worse.

Experiment	%Miss	%FA	%Spkr	%Total
SDM, SAD_{org}	5.30	2.30	11.50	19.07
SDM, SAD_{cvn}	5.10	2.10	12.80	19.93
SDM, SAD_{noise}	6.50	1.70	10.60	18.74
MDM, SAD_{org}	4.60	2.00	6.30	12.90
MDM, SAD_{cvn}	4.40	1.90	7.30	13.60
MDM, SAD_{noise}	5.00	1.60	8.00	14.68

Table 2: *The results of experiments with the original SAD component (SAD_{org}), the SAD component with cepstrum variance normalization and flooring (SAD_{cvn}) and the component without noise reduction (SAD_{noise}) on the SDM and MDM conditions.*

3.2. Delay feature stream

The delay features are generated as by-product of the beamforming of the audio. In order to determine the optimum window size for calculating these features, we tried three window sizes: 64 ms, 250 ms and 500 ms. We tested the three configurations on all three SAD configuration. The results are listed in table 3.

Experiment	64 ms %DER	250 ms %DER	500 ms %DER
SAD_{org}	12.21	13.19	12.90
SAD_{cvn}	14.13	12.66	13.60
SAD_{noise}	14.91	13.00	14.68

Table 3: *The results of experiments with the three window lengths for delay-feature calculation on our three SAD configurations*

4. Combining system outputs

In section 2 we have discussed our RT09 speaker diarization system and in the previous section we have described a number of system configurations that we created to determine the best set-up of our system. In this section we will describe our approach to combine the output of these system configurations into one single output, but first we will discuss two other interesting studies in which multiple diarization systems are combined.

4.1. Existing methods

In [11], a 'fusion' system is created out of two input systems in three steps. First, new clusters are created from the input segmentations in a way that each new cluster contains only speech from one single cluster of both input segmentations. This can easily be done by simply concatenating the cluster IDs from the input segmentations into one new cluster (speaker 01 from the first segmentation and speaker A from the second form speaker 01-A etc). Second, all new clusters that have a large amount of data assigned are considered correct and passed to the output. In the third step, all other clusters (except for the very small ones that are simply deleted) are re-evaluated by one of the original systems. This approach is interesting, but has two disadvantages. First, it requires the use of a single diarization system in the final step. As long as there is not a single system that outperforms all others, the question remains which system to use as final judge. Second, if one of the systems makes the mistake of cutting up speech from a single speaker (under-clustering), using this method it is not possible to recover from this mistake.

In [12], the output of two systems are merged in a slightly different manner. First, new clusters are created in the same way as in [11] (SPKA + SPKB = SPKAB), but only the clusters where *all* speech from the first input cluster and *all* speech from the other form a single new cluster are passed directly to the output. All other clusters are put in so called 'supergroups' so that each group consists of a number of clusters that is not overlapping with any of the other groups. Next, for each supergroup the output clusters are formed by selecting the best combinations of clusters in the supergroup. Different metrics can be used as a selection criterion, but in [12] the Bayesian Information Criterion (BIC) applied on GMMs works the best. Impressively, for some recordings this merged system outperforms both input systems. There are two disadvantages to using this method though. First, the method used to generate the final clusters (modeling the clusters and apply BIC) is the same as we use in our diarization system and may make the same mistakes. Similar as the method in [11], using this method, we still have to decide on the optimal final diarization set-up. A second problem of this approach is that when combining more than two systems, the computational complexity becomes high as a lot of possible cluster combinations needs to be compared.

The previous two methods both attempt to generate a segmentation that is even *better* than all of the input segmentations. In our approach, from a list of input segmentations we will simply try to select one of the better ones.

4.2. Our approach

The errors that our system can make that affect diarization error rate the most are to merge models of two different speakers and to stop clustering too early or too late. If one of these mistakes is made, the DER for that particular recording often increases with more than twenty percent. Therefore, although we would be interested in generating a segmentation that is *better* than all input segmentations, for now we will focus on selecting a segmentation output in which these kinds of destructive errors have not been made.

As we have seen in our experiments, sometimes the system will or will not make one of the more destructive errors just because of a small system adjustment. Applying CVN in the SAD component for example, will influence the overall DER only a little (almost one percent for the SDM condition), but for individual recordings the DER can increase or decrease considerably (for SDM, one meeting goes from 32% to 49% while another one goes from 21% to 5%). We assume that the system makes these kinds of mistakes after a tuning adjustments because it is especially difficult to distinct between two particular speakers. If we change something else in the system, the same mistake might be made but we assume that no other big error is introduced. We also assume that in the majority of the cases the system will *not* make this particular mistake. If these assumptions are correct, we can improve the overall DER by applying voting by majority and follow this procedure:

- Determine the distance between each pair of input segmentations.
- Using these distances, apply agglomerative clustering until 2 groups of segmentations are left.
- From the biggest group, pick the one that has the smallest distance to all other segmentations in the group.

If it is true that if a big error is made during segmentation it is always the same one, we know that all segmentations in which the error is not made will be very similar and also the segmentations in which the error *is* made will be very similar. Therefore, if we have a metric to measure similarity (or distance) we can divide the segmentations into two groups: good and bad[2]. Because we assume that the good group is bigger than the bad group and we know that the segmentations in this group are very similar, we chose the segmentation that is in the center of the group: the one with the smallest distance to all other segmentations.

As a distance measure we simply use a symmetric diarization error rate. For two segmentations A and B, we measure the DER of A with B as reference and add that score to the DER of B with A as reference.

5. Experiments

In this section we will discuss the experiments that we have performed using the majority voting procedure as we described in the previous section. For each experiment we use a combination of the system outputs discussed in section 3. All diarization error rates are calculated on the set of conference meetings listed in table 1.

5.1. Voting for the best SDM system

For the SDM condition we used the diarization system with the three different SAD configurations as input for our voting scheme. In table 4, the DER of each of these system configurations is listed together with the DER of the combined system output (using our voting approach) and the best and worst possible DER if our algorithm would have picked the best or worst DER from the input segmentations each time.

Experiment	%DER
SAD_{org}	19.07
SAD_{cvn}	19.93
SAD_{noise}	18.74
Worst possible combination	23.37
Best possible combination	15.94
Voted combination	**17.27**

Table 4: *The results of our SDM voting experiment for the three SAD configurations.*

[2]with three groups, we had have to introduce 'the ugly' as well...

As can be seen in table 4, the voted combination of system outputs is better than each of the input segmentations. A comparison with the best possible combination of input segmentations shows that the voting method did not pick the optimal segmentation. A theoretical improvement of the voting method of 1.33% DER absolute is possible.

5.2. Voting for the best MDM system

For the multiple distant microphone condition we have performed two sets of experiments. First, for each of the three delay feature window lengths we have used our three SAD configurations (SAD_{org}, SAD_{cvn} and SAD_{noise}) as input for our voting algorithm (these configurations are all described in section 3). Second, we used all nine configurations together as input for the voting algorithm.

The results of these two sets of experiments are listed in table 5. As can be seen from the table, the voted output is not always better than the best input segmentation (for example for the 64 ms configuration), but it is always better than the average of the inputs. Apparently, similar to ROVER for ASR, in order to get an improvement it is important that the input segmentations are all of comparable quality.

Further it can be seen in table 5 that the voted output of the final experiment is 1.4% DER absolute worse than the best possible combination of input segmentations and 6.9% DER absolute better than the worst possible combination.

Experiment	64 ms %DER	250 ms %DER	500 ms %DER
SAD_{org}	12.21	13.19	12.90
SAD_{cvn}	14.13	12.66	13.60
SAD_{noise}	14.91	13.00	14.68
Worst possible combinations	16.63	14.31	16.27
Best possible combinations	11.55	11.62	11.45
Voted combinations	**12.33**	**12.92**	**12.89**
Worst combination of all		18.66	
Best combination of all		10.39	
Voted combination of all		**11.77**	

Table 5: *The results of our MDM voting experiments. We first combine the outputs of each SAD configuration for each delay window length and then we combine all nine systems into one single output.*

6. Discussion

In this paper we presented our method for combining multiple diarization systems into one single system by applying a majority voting scheme. The system configurations that we used as input for voting were all configurations that we tested during the fine-tuning of our system. The goal of applying the voting scheme was to make the system less sensitive to the parameters that we changed during fine-tuning. For both our SDM and MDM systems the voting method improved the diarization error rate.

Although the overall DER improves, our voting method is not able to improve the results of individual meetings. In future work we will investigate the possibility of combining our method with other merging methods that *are* able to do so. For example, our method could be used to select the two best systems after which it is possible to use combination techniques such as in [11, 12].

Running multiple configurations of our diarization system in parallel is time consuming and might not be feasible for processing entire archives. Therefore, in future work we will investigate if we can use this technique to improve fast, but less accurate diarization systems and use information from the output (for example high confident clusters or the number of speakers) in a final, more accurate diarization run.

7. Acknowledgments

This paper is partly based on research carried out in the European Union 6th FWP IST Integrated Project AMIDA (Augmented Multi-party Interaction with Distant Access, FP6-506811) and the project CHoral - Access to Oral History, which is funded by the NWO program CATCH (`http://www.nwo.nl/catch`).

8. References

[1] D. Reynolds and P. Torres-Carrasquillo, "Approaches and applications of audio diarization," Philadelphia, PA, March 2005, pp. 953–956.

[2] J. G. Fiscus, J. Ajot, and J. S. Garofolo, "The rich transcription 2007 meeting recognition evaluation," in *Multimodal Technologies for Perception of Humans*, ser. Lecture Notes in Computer Science. Berlin: Springer Verlag, 2008.

[3] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (rover)," in *proceedings 1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, CA, 1997, pp. 347–352.

[4] J. Ajmera, H. Bourlard, I. Lapidot, and I. McCowan, "Unknown-multiple speaker clustering using HMM," in *proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Denver, Colorado, USA, 2002.

[5] M. Huijbregts, C. Wooters, and R. Ordelman, "Filtering the unknown: Speech activity detection in heterogeneous video collections," in *proceedings of Interspeech*, Antwerp, Belgium, August 2007.

[6] D. van Leeuwen and M. Huijbregts, "The AMI speaker diarization system for NIST RT06s meeting data," in *Machine Learning for Multimodal Interaction (MLMI)*, ser. Lecture Notes in Computer Science, vol. 4299. Berlin: Springer Verlag, October 2007, pp. 371–384.

[7] A. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grezl, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, and S. Sivadas, "Qualcomm-icsi-ogi features for asr," in *proceedings of ICSLP*, 2002.

[8] X. Anguera, "Robust speaker diarization for meetings," Ph.D. dissertation, Universitat Politecnica De Catalunya, 2006.

[9] J. M. Pardo1, X. Anguera, and C. Wooters, "Speaker diarization for multiple distant microphone meetings: Mixing acoustic features and inter-channel time differences," in *proceedings of Interspeech*, 2006.

[10] C. Wooters and M. Huijbregts, "The ICSI RT07s speaker diarization system," in *Multimodal Technologies for Perception of Humans*, ser. Lecture Notes in Computer Science. Berlin: Springer Verlag, 2008.

[11] D. Moraru, S. Meignier, C. Fredouille, L. Besacier, and L. Bonastre, "The elisa consortium approaches in broadcast news speaker segmentation during the nist 2003 rich transcription evaluation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004.

[12] S. E. Tranter, "Two-way cluster voting to improve speaker diarisation performance," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.